

ĐẠI HỌC QUỐC GIA TP HCM
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC HÀNH



LAB: 4

LẬP LỊCH TIẾN TRÌNH

Lớp thực hành môn: **HỆ ĐIỀU HÀNH**

– **IT007.M11.CLC.1**



Sinh viên thực hiện:

MSSV: 20520322

Họ và tên: Nguyễn Thị Mỹ Trân



4.4 Hướng dẫn thực hành

Soạn thảo và biên dịch giải thuật FCFS

Viết chương trình lưu vào file: NguyenThiMyTran_20520322_LAB4_FIFO

```
#include<stdio.h>

void main(){

    int pn[10];

    int arr[10], bur[10], star[10], finish[10], tat[10], wt[10], i, n;

    double totwt=0, tottat=0;

    printf("Enter the number of processes:");

    scanf("%d",&n);

    for(i=0;i<n;i++) {

        printf("Enter the Process Name, Arrival Time & Burst Time:");

        scanf("%d%d%d",&pn[i],&arr[i],&bur[i]);

    }

    for(i=0;i<n;i++) {

        if(i==0) {

            star[i]=arr[i];

            wt[i]=star[i]-arr[i];

            finish[i]=star[i]+bur[i];

            tat[i]=finish[i]-arr[i];

        } else{

            star[i]=finish[i-1];

            wt[i]=star[i]-arr[i];

            finish[i]=star[i]+bur[i];

            tat[i]=finish[i]-arr[i];

        }

    }
```



```
}

printf("\nPName Arrtime Burttime Start TAT Finish");

for(i=0;i<n;i++) {

printf("\n%d\t%d\t\t%d\t\t%d\t\t%d",pn[i],arr[i],bur[i],star
[i],tat[i],finish[i]);

totwt+=wt[i];

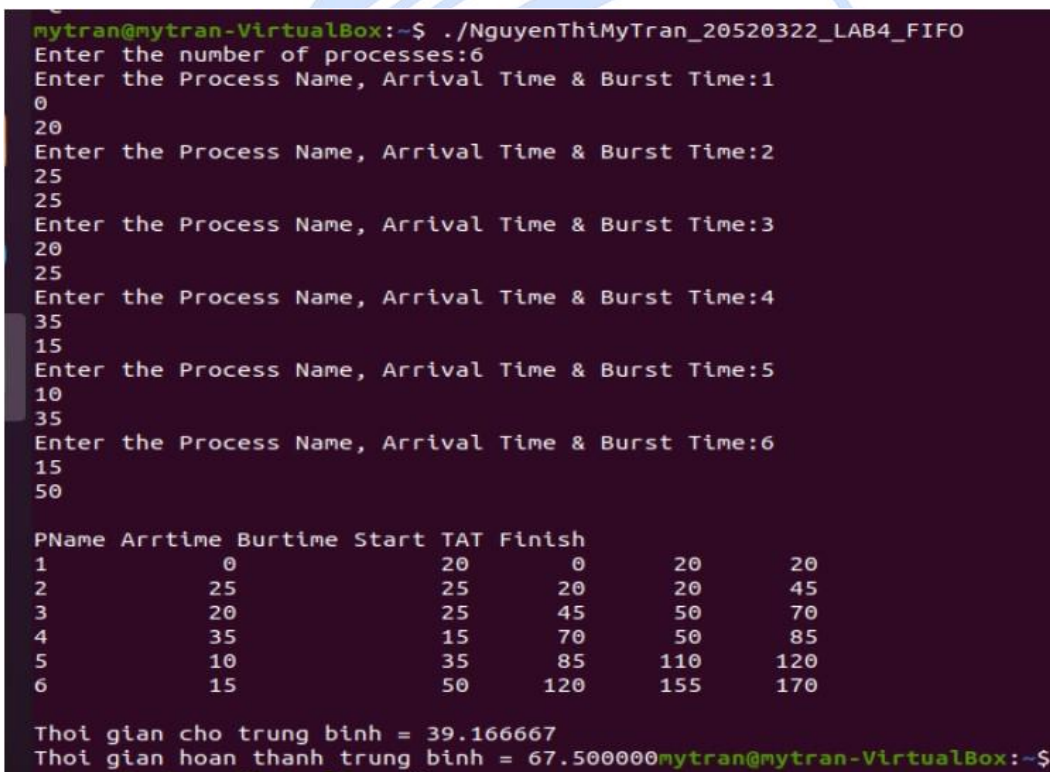
tottat+=tat[i];

}

printf("\n\nThoi gian cho trung binh = %lf\n",totwt/n);

printf("Thoi gian hoan thanh trung binh = %lf",tottat/n);
}
```

Kết quả chạy chương trình:



```
mytran@mytran-VirtualBox:~$ ./NguyenThiMyTran_20520322_LAB4_FIFO
Enter the number of processes:6
Enter the Process Name, Arrival Time & Burst Time:1
0
20
Enter the Process Name, Arrival Time & Burst Time:2
25
25
Enter the Process Name, Arrival Time & Burst Time:3
20
25
Enter the Process Name, Arrival Time & Burst Time:4
35
15
Enter the Process Name, Arrival Time & Burst Time:5
10
35
Enter the Process Name, Arrival Time & Burst Time:6
15
50

PName Arrtime Burttime Start TAT Finish
1      0          20      0      20      20
2      25          25      20      20      45
3      20          25      45      50      70
4      35          15      70      50      85
5      10          35      85      110     120
6      15          50      120     155     170

Thoi gian cho trung binh = 39.166667
Thoi gian hoan thanh trung binh = 67.500000mytran@mytran-VirtualBox:~$
```

4.5 Bài tập ôn tập

1. Viết chương trình mô phỏng giải thuật SJF với các yêu cầu sau:

❖ Nhập số lượng process



❖ **Nhập process name, arrival time, burst time**

❖ **In ra Process name, response time, waiting time, turnaround time, average waiting time, average turnaround time**

Viết chương trình lưu tại file: NguyenThiMyTran_20520322_LAB4_Bai1.c

```
#include <stdio.h>

// Ham swap hai gia tri a va b
void swap (int *a ,int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Xoa a[cu] va them vao vi tri moi trong mang a[moi]
void deleteThenInsert(int a[], int n, int cu, int moi)
{
    int temp = a[cu];
    if(cu < 0 || moi < 0 || cu >=n || moi >= n || moi == cu)
        return;
    else if(moi < cu)
    {
        for(int i = cu; i > moi; i--)
            a[i] = a[i-1];
        a[moi] = temp;
    }
}
```



```
}

else

{

    for(int i = cu; i < moi; i++)

        a[i] = a[i+1];

    a[moi] = temp;

}

}

// Vi tri i dau tien trong mang a ma a[i] nho nhat (start <= i <= end)
int minAtIndex(int a[], int n, int start, int end)
{
    if( start > end || start < 0 || end >= n)

        return -1;

    int index = start;

    for(int i = start + 1; i <= end; i++)

        if(a[i] < a[index])

            index = i;

    return index;
}

void main(){

    /* arr for arrival time, bur for burst time, star for start
    time, finish for finish time, tat for turnaround time, wt for waiting
    time, average wating time and average turnaround time*/

    int pn[20], arr[20], bur[20], star[20], finish[20], tat[20],
    wt[20], i, j, k, n;

    int totwt = 0, tottat = 0;

    float awt, atat;

    // Nhap so process, ten process, arrival time, burst time
    printf("Enter the number of processes: ");
```



```
scanf("%d", &n);

for(i = 0; i < n; i++) {
    printf("Enter the Process Name, Arrival Time & Burst
Time: ");

    scanf("%d%d%d", &pn[i], &arr[i], &bur[i]);
}

// Sắp xếp lại các process theo arrival time tăng dần
for(i = 0; i < n - 1; i++)
    for(j = i + 1; j < n; j++)
        if(arr[i] > arr[j])
        {
            swap(&arr[i], &arr[j]);
            swap(&bur[i], &bur[j]);
            swap(&pn[i], &pn[j]);
        }

// Tìm process có burst time và arrival time nhỏ nhất
// Tìm vị trí i sao cho arr[i] > arr[0]
i = 1;
while(i < n && arr[i] == arr[0])
    i++;

// vị trí của pn thỏa bur bé nhất trong [0, i]
j = minAtIndex(bur, n, 0, i - 1);

// Đưa process thứ j lên chạy đầu tiên
if(j != 0)
{
    deleteThenInsert(arr, n, j, 0);
```



```
        deleteThenInsert(bur, n, j, 0);

        deleteThenInsert(pn, n, j, 0);

    }

    /* Calculate start time, waiting time, finish time, turnaround
time của moi process */

    // Process chạy đầu tiên

    star[0] = arr[0];

    wt[0] = 0;

    finish[0] = star[0] + bur[0];

    tat[0] = finish[0] - arr[0];


    // Process chạy từ thứ 2 trở đi

    for(k = 1; k < n; k++) {

        if(arr[k] > finish[k - 1])

        {

            i = k + 1;

            while(i < n && arr[i] == arr[k])

                i++;

            // vị trí của pn thỏa bur bé nhất

            j = minAtIndex(bur, n, k, i - 1);

            if(j != k)

            {

                deleteThenInsert(arr, n, j, k);

                deleteThenInsert(bur, n, j, k);

                deleteThenInsert(pn, n, j, k);

            }

            star[k] = arr[k];

        }

        else
```



```
{

    i = k + 1;

    while(i < n && arr[i] <= finish[k - 1])

        i++;

    j = minAtIndex(bur, n, k, i - 1);

    if(j != k)

    {

        deleteThenInsert(arr, n, j, k);

        deleteThenInsert(bur, n, j, k);

        deleteThenInsert(pn, n, j, k);

    }

    star[k] = finish[k-1];

}

finish[k] = star[k] + bur[k];

wt[k] = star[k] - arr[k];

tat[k] = finish[k] - star[k];

}

// In ra màn hình

printf("\nPName\tArctime\tBurtime\tStart\tFinish\tTAT\tWT");

for(i = 0; i < n; i++) {

    printf("\n%5d\t%7d\t%7d\t%5d\t%6d\t%3d\t%2d", pn[i],
arr[i], bur[i], star[i], finish[i], tat[i], wt[i]);

    // total waiting time va total turnaround time

    totwt += wt[i];

    tottat += tat[i];

}

// Calculate and Print average waiting time va average turnaround
time

awt = (float)totwt / n;
```




```
atat = (float)tottat / n;
```

```
printf("\n\nAverage waiting time la: %f\n", awt);
```

```
printf("Average turnaround time la: %f\n", atat);
```

```
}
```

Kết quả chạy chương trình:

Process	Arrival time(AT)	Burst time(BT)
P1	0	20
P2	25	25
P3	20	25
P4	35	15
P5	10	35
P6	15	50

```
mytran@mytran-VirtualBox:~$ gcc NguyenThiMyTran_20520322_LAB4_Bai1.c -o NguyenThiMyTran_20520322_LAB4_Bai1
mytran@mytran-VirtualBox:~$ ./NguyenThiMyTran_20520322_LAB4_Bai1
Enter the number of processes: 6
Enter the Process Name, Arrival Time & Burst Time: 1
0
20
Enter the Process Name, Arrival Time & Burst Time: 2
25
25
Enter the Process Name, Arrival Time & Burst Time: 3
20
25
Enter the Process Name, Arrival Time & Burst Time: 4
35
15
Enter the Process Name, Arrival Time & Burst Time: 5
10
35
Enter the Process Name, Arrival Time & Burst Time: 6
15
50

PName  Arrtime  Burtime  Start   Finish   TAT     WT
1       0        20      0       20       20      0
3       20       25      20      45       25      0
4       35       15      45      60       15      10
2       25       25      60      85       25      35
5       10       35      85      120      35      75
6       15       50     120     170      50     105

Average waiting time la: 37.500000
Average turnaround time la: 28.333334
mytran@mytran-VirtualBox:~$
```



2. Viết chương trình mô phỏng giải thuật SRT với các yêu cầu sau:

❖ Nhập số lượng process

❖ Nhập process name, arrival time, burst time

❖ In ra Process name, response time, waiting time, turnaround time, average waiting time, average turnaround time

Viết chương trình lưu tại file: NguyenThiMyTran_20520322_LAB4_Bai2.c

```
#include<stdio.h>

void main()
{
    int a[10],b[10],x[10];
    int waiting[10],turnaround[10],completion[10];
    int i,j,smallest,count=0,time,n;
    double avg=0,tt=0,end;

    printf("\nEnter the number of Processes: ");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("\nEnter arrival,burst time of process %d : ",i+1);
        scanf("%d",&a[i]);
        scanf("%d",&b[i]);
    }
}
```



```
}

for(i=0;i<n;i++)
x[i]=b[i];

b[9]=9999;
//printf("time => process number");
for(time=0;count!=n;time++)
{
    smallest=9;
    for(i=0;i<n;i++)
    {
        if(a[i]<=time && b[i]<b[smallest] && b[i]>0 )
            smallest=i;
    }
    b[smallest]--;
    //printf("\n%d => p%d",time+1,smallest);
    if(b[smallest]==0)
    {
        count++;
        end=time+1;
        completion[smallest] = end;
        waiting[smallest] = end - a[smallest] - x[smallest];
        turnaround[smallest] = end - a[smallest];
        // printf("\n %d  %d  %d",smallest,wt[smallest],ttp[smallest]);
    }
}

printf("pid \t burst \t arrival \twaiting \tturnaround \tcompletion");
for(i=0;i<n;i++)
```



```
{  
    printf("\n %d \t    %d \t %d\t\t%d  
\t\t%d\t\t%d",i+1,x[i],a[i],waiting[i],turnaround[i],completion[i]);  
  
    avg = avg + waiting[i];  
  
    tt = tt + turnaround[i];  
  
}
```



```
printf("\n\nThoi gian cho trung binh = %lf\n",avg/n);  
  
printf("Thoi gian hoan thanh trung binh = %lf",tt/n);  
  
}
```

Kết quả chạy chương trình:

Process	Arrival time(AT)	Burst time(BT)
P1	0	20
P2	25	25
P3	20	25
P4	35	15
P5	10	35
P6	15	50

```
mytran@mytran-VirtualBox:~$ gcc NguyenThiMyTran_20520322_LAB4_Bai2.c -o NguyenThiMyTran_20520322_LAB4_Bai2  
mytran@mytran-VirtualBox:~$ ./NguyenThiMyTran_20520322_LAB4_Bai2  
  
Enter the number of Processes: 6  
Enter arrival,burst time of process 1 : 0  
20  
Enter arrival,burst time of process 2 : 25  
25  
Enter arrival,burst time of process 3 : 20  
25  
Enter arrival,burst time of process 4 : 35  
15  
Enter arrival,burst time of process 5 : 10  
35  
Enter arrival,burst time of process 6 : 15  
50  
pid      burst  arrival    waiting    turnaround  completion  
1         20     0           0           20           20  
2         25     25          35          60           85  
3         25     20           0           25           45  
4         15     35          10          25           60  
5         35     10          75          110          120  
6         50     15          105         155          170  
  
Thoi gian cho trung binh = 37.500000  
Thoi gian hoan thanh trung binh = 65.833333mytran@mytran-VirtualBox:~$  
mytran@mytran-VirtualBox:~$
```



3. Viết chương trình mô phỏng giải thuật RR với các yêu cầu sau (giả sử tất cả các tiến trình đều có arrival time là 0):

❖ Nhập số process

❖ Nhập quantum time

❖ Nhập process name, burst time

❖ In ra Gantt chart với các thông số: process name, start processor time, stop processor time

❖ In ra average waiting time và average turnaround time

Viết chương trình lưu tại file: NguyenThiMyTran_20520322_LAB4_Bai3.c

```
#include<stdio.h>

int main()
{

    int count,j,n,time,remain,flag=0,time_quantum;
    int wait_time=0,turnaround_time=0,at[10],bt[10],rt[10];
    printf("Enter Total Process:\t ");
    scanf("%d",&n);
    remain=n;
    for(count=0;count<n;count++)
```



```
{

    printf("Enter Burst Time for Process Process Number %d :",count+1);

    at[count]=0;

    scanf("%d",&bt[count]);

    rt[count]=bt[count];

}

printf("Enter Time Quantum:\t");

scanf("%d",&time_quantum);

printf("\n\nProcess\t|stop Time|start Time\n\n");

for(time=0,count=0;remain!=0;)

{

    if(rt[count]<=time_quantum && rt[count]>0)

    {

        int starttime;

        starttime = time;

        time+=rt[count];

        rt[count]=0;

        flag=1;

        printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-
at[count],starttime);

    }

    else if(rt[count]>0)

    {

        rt[count]-=time_quantum;

        time+=time_quantum;

        printf("P[%d]\t|\t%d\t|\t%d\n",count+1,time-at[count],time-
time_quantum);

    }

    if(rt[count]==0 && flag==1)
```



```
{  
    remain--;  
  
    wait_time+=time-at[count]-bt[count];  
    turnaround_time+=time-at[count];  
    flag=0;  
}  
if(count==n-1)  
    count=0;  
else if(at[count+1]<=time)  
    count++;  
else  
    count=0;  
}  
  
printf("\nThoi gian cho trung binh %f\n",wait_time*1.0/n);  
printf("Thoi gian hoan thanh trung binh = %f",turnaround_time*1.0/n);  
  
return 0;  
}
```

Kết quả chạy chương trình:

Process	Arrival time(AT)	Burst time(BT)
P1	0	20
P2	25	25
P3	20	25
P4	35	15
P5	10	35
P6	15	50



```
mytran@mytran-VirtualBox:~$ gcc NguyenThiMyTran_20520322_LAB4_Bai3.c -o NguyenThiMyTran_20520322_LAB4_Bai3
mytran@mytran-VirtualBox:~$ ./NguyenThiMyTran_20520322_LAB4_Bai3
Enter Total Process: 6
Enter Burst Time for Process Process Number 1 :20
Enter Burst Time for Process Process Number 2 :25
Enter Burst Time for Process Process Number 3 :25
Enter Burst Time for Process Process Number 4 :15
Enter Burst Time for Process Process Number 5 :35
Enter Burst Time for Process Process Number 6 :50
Enter Time Quantum: 10

Process |stop Time|start Time
P[1]    |    10    |    0
P[2]    |    20    |   10
P[3]    |    30    |   20
P[4]    |    40    |   30
P[5]    |    50    |   40
P[6]    |    60    |   50
P[1]    |    70    |   60
P[2]    |    80    |   70
P[3]    |    90    |   80
P[4]    |   95    |   90
P[5]    |   105    |   95
P[6]    |   115    |  105
P[2]    |   120    |  115
P[3]    |   125    |  120
P[5]    |   135    |  125
P[6]    |   145    |  135
P[5]    |   150    |  145
P[6]    |   160    |  150
P[6]    |   170    |  160

Thời gian cho trung bình 93.333333
Thời gian hoàn thành trung bình = 121.666667mytran@mytran-VirtualBox:~$
mytran@mytran-VirtualBox:~$
```

HẾT